(2)

NPS-AA-93-001

NAVAL POSTGRADUATE SCHOOL

Monterey, California

A RAPID COMPUTATIONAL MODEL
FOR ESTIMATING THE PERFORMANCE OF
COMPLIANT AIRFOILS IN CASCADES

by

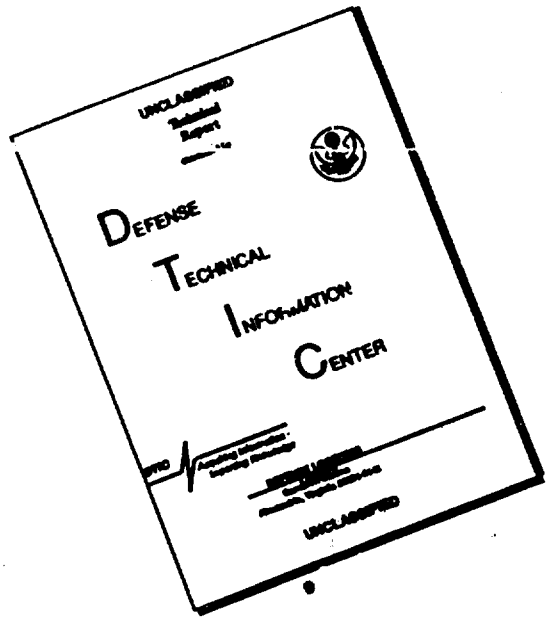
F. Sisto
M. Avila

Technical Report for Period
January 1992 - June 1992

Approved for public release; distribution unlimited

Prepared for: Naval Postgraduate School
Monterey, CA 93943

DISCLAIMER NOTICE



THIS DOCUMENT IS BEST QUALITY AVAILABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

NAVAL POSTGRADUATE SHOOL
MONTEREY, CA 93943

Rear Admiral T. A. Mercer
Superintendent

Harrison Shull
Provost

This report was prepared in conjunction with research conducted for the Naval Postgraduate School and funded by the Naval Postgraduate School.

This report was prepared by:

F. SISTO
Senior Research Associate

for M. F. Platzer
M. AVILA
LT, U.S. Navy

Reviewed by:

M. F. Platzer
M. F. PLATZER
Director, Navy-NASA Joint Institute

Released by:

P. J. Marto
PAUL J. MARTO
Dean of Research

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) NPS-AA-93-001	5 MONITORING ORGANIZATION REPORT NUMBER(S) NPS-AA-93-001
6a. NAME OF PERFORMING ORGANIZATION Naval Postgraduate School	6b OFFICE SYMBOL (If applicable) AA
7a NAME OF MONITORING ORGANIZATION Naval Postgraduate School	
6c. ADDRESS (City, State, and ZIP Code) Monterey, CA 93943	7b ADDRESS (City, State, and ZIP Code) Monterey, CA 93943
8a. NAME OF FUNDING/SPONSORING ORGANIZATION Naval Postgraduate School	8b OFFICE SYMBOL (If applicable) AA
9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER National Research Council	
8c. ADDRESS (City, State, and ZIP Code) Monterey, CA 93943	10 SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO	PROJECT NO	TASK NO	WORK UNIT ACCESSION NO

11. TITLE (Include Security Classification)

A Rapid Computational Model for Estimating the Performance of Compliant Airfoils in Cascades

12. PERSONAL AUTHOR(S)

F. Sisto and M. Avila

13a. TYPE OF REPORT	13b TIME COVERED	14 DATE OF REPORT (Year, Month, Day)	15 PAGE COUNT
Technical	FROM 1/92 TO 6/92	92/07/01	29

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Turbomachinery Aeroelasticity

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

We consider the problem of designing the blades in a turbomachine to have a specific schedule of structural stiffness in order to achieve more favorable aerothermodynamic performance characteristics than is possible with rigid blades. This problem of "aero-thermodynamic tailoring" is extremely complex because of the need to couple the flow computation over a deformable surface with the structural dynamics computation. In this report an analysis is presented which accomplishes this objective by means of a two-dimensional incompressible potential flow model using vortex sheet representation of the blades. Details of the analysis and of the resulting computer program together with representative results are described.

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT	21 ABSTRACT SECURITY CLASSIFICATION
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS	UNCLASSIFIED

22a NAME OF RESPONSIBLE INDIVIDUAL	22b TELEPHONE (Include Area Code)	22c OFFICE SYMBOL
M. F. Platzer	(408)656-2058	AA/Pl

DD FORM 1473, 84 MAR 83 APR edition may be used until exhausted
All other editions are obsolete

U.S. Government Printing Office 1986—606 24.

A RAPID COMPUTATIONAL MODEL FOR ESTIMATING THE PERFORMANCE OF COMPLIANT AIRFOILS IN CASCADES

F. Sisto and M. Avila

SUMMARY

By designing the blades in a turbomachine to have a specific schedule of structural stiffness (typically more compliant than normal) it is possible to endow the end product with an aerothermodynamic performance characteristic that is different from the rigid-bladed counterpart. Presumably the resulting characteristic of the compliant turbomachine, within certain operational limits, is preferable to the more rigid turbomachine. Since there is no active controller involved in changing the blade geometry in response to changing operating conditions, the process may be thought of as passive control by aeroelastic tailoring. In fact, since the objective function is a change in the performance characteristic, the overarching process may be termed "aerothermoelastic tailoring": the principles of static aeroelasticity are applied to a turbomachine to change its performance in a thermodynamic sense.

The problem posed is extremely complex stemming from the need to couple the flow computation over a deformable surface with the structural dynamics computation of the body occupying the other side of the internal boundary. If no aeroelastic instabilities are encountered at a particular operating point the resulting configuration is fixed in time and the static solution (flow and structure) is a special case of the more general dynamic formulation.

In this situation it is useful to develop an approximate model for use in the rapid exploration of many static aeroelastic configurations, described by a host of controlling parameters. In this fashion those regions of parameter space showing potential advantages for the compliant-bladed turbomachine may then be computationally refined using more accurate, but more time-consuming, CFD coupled to discretized structural dynamic codes (FEM). It is the purpose of the present model to permit this rapid exploration of parameter space to delineate the more promising regions for the subsequent detailed studies.

I. AERODYNAMICS

For the rapid computation of the aerodynamic loads acting on airfoils in cascade, a two-dimensional incompressible potential flow model is chosen using a vortex sheet representation of the thin airfoils. Under these assumptions the upwash induced along the chord of the reference (zeroth) airfoil is given by

$$v(x) = \frac{1}{2\pi} \oint_0^1 g(\xi) \frac{d\xi}{\xi - x} + \frac{1}{2\pi} \int_0^1 g(\xi) F(\xi - x)\, d\xi \qquad (1)$$

Here the variables x(and ξ) are chordwise coordinates made dimensionless w.r.t. the chord, c, and g(x) is the unknown strength of the vortex sheet representing each airfoil. The first integral represents the contribution of the reference airfoil and the second integral represents the contribution from the remainder of the cascade through the "influence function" F under the assumption that every blade has the identical distribution g(x).

Figure 1 shows the geometric configuration under analysis. Eq (1) is derived in Appendix A, as well as the inversion of the integral equation for the solution g(x) under appropriate non-penetration boundary conditions and specified vector mean velocity of flow (V,α).

The solution is obtained under the classical thin airfoil coordinate transformation

$$x = \tfrac{1}{2}(1 - \cos\theta) \qquad (2)$$

$$\theta = \cos^{-1}(1 - 2x) \qquad (2')$$

followed by a discretization of the transformed variable

$$\theta = \frac{\ell\pi}{n}, \quad 0 \le \ell \le n \qquad (3)$$

where n is the number of segments along the chord, or n+1 is the number of gridpoints at which the boundary conditions are satisfied and also the values of $g(\theta)$ are determined.

Using results first obtained in (Sisto, 1967) the distributed circulation can be expressed in the following form

$$g(x) = V\sqrt{\frac{1-x}{x}}\, f(x) \qquad (4)$$

(thus rendering f dimensionless) and the solution for f in matrix format appears as follows:

$$\{f(\theta)\} = [A_{ij}]^{-1}\{U_j\} \qquad (5)$$

Here A is the (n+1)x(n+1) aerodynamic matrix and U is the upwash column matrix or vector.

The expression for the upwash U depends on the mean flow angle α and y' the camberline slope distribution. The expression for A is quite complex. It reflects the geometry of the cascade: s/c, β and, again, y' as a result of the method of inverting the fundamental integral equation. The relevant expressions appear as Eqs. (A2) (A14) and (A16) determining A and Eqs. (A14) (A19) (A20) and (A21) for U. Both have been programmed for solution in the Matlab language and form part of the program test00.m and solve.m appearing as Appendix C.

II. STRUCTURE

Each of the identical airfoil sections comprising the cascade are modelled as two-dimensional plates of small curvature and variable thickness. Each plate may be thought of as the characteristic or representative section of an actual blade at 75 or 80 percent of the span of a typical cantilever blade. As such the elastic displacements may be expressed in terms of the bending moment, M according to

$$\frac{d^2y}{dx^2} = \frac{M(x)}{E(x)} \tag{6}$$

where E is the local stiffness related to Young's modulus, cross-sectional moment-of-inertia and Poisson's ratio if a solid orthotropic material is under consideration. For an aeroelastically tailored structure it is presumed that E is under the control of the designer within limits determined by strength considerations, stability, etc.

Recognizing that changes in stagger related to the untwist of a three-dimensional blade may be an important aeroelastic parameter, the two-dimensional counterpart model will be supported at the leading-edge and near the trailing edge by springs whose stiffness may be specified. In actual construction of a compliant blade these springs represent the restraint to bending provided by the spanwise spars required to support the more compliant material comprising the airfoil shaped profile.

The elastic bending moment M(x) is therefore a function of the aerodynamic loading whereas the elastic displacements depend as well upon the strength of leading edge and rearward springs. The structural configuration is sketched in Fig. 2 where only a single airfoil need be shown, the others being identical. (The case of a shrouded cascade of airfoils would need additional modification of the model).

The aeroelastic coupling is provided by this aerodynamic loading acting on the compliant structure. (There may be a small amount of camber in the unstrained state upon which the elastic displacements are superposed). The aerodynamic load is provided by

$$dL = \zeta V d\Gamma = \zeta V g(x)\, d(cx)$$
$$dL = \zeta V^2 \sqrt{\frac{1-x}{x}}\, f(x)\, d(cx) \qquad (7)$$

It is this loading which is applied to the plate; the details are developed in Appendix B. It may be noticed that dL has a square root singularity at the leading edge. However, under the coordinate transformation given by Eq (2), the singularity is integrable in the determination of lift and moment, viz

$$dL = qc(1+\cos\theta)\, f(\theta)\, d\theta \qquad (7')$$

where

$$q = \tfrac{1}{2}\zeta V^2 \qquad (8)$$

is the dynamic presssure based on the vector mean velocity, V. For known aerodynamic loading and structural stiffness, Eq (6) may be solved for the elastic displacements Y(x). Subsequently the unstrained slopes may be augmented by the elastic slopes to obtain the overall slope distribution. This total slope function comprises the input to the aerodynamics subroutine for computation of the aerodynamic loading.

III. COMPUTATIONAL LOGIC

Conceptually the problem at hand is an eigenvalue problem similar in concept to the problem of wing divergence. However, owing to the complexity of the aerodynamic formulation, the method of solution is by iteration. An assumed blade shape produces a specific aerodynamic load. This load applied to the blade structure produces a specific schedule of displacements. These displacements define a new blade shape and completes the first loop of the iteration. If the iterations converge the stable static solution is obtained. If the iterations do not converge the condition of "divergence" states that the initial set of aerodynamic and structural parameters are in supercritical region of parameter space and no stable solution is obtainable.

The looping concept is illustrated in Fig. 3, a flow chart of the computational logic used to iterate the solution.

This chart, along with the actual program statements (expressed in MatLab language) comprise Appendix C.

IV. PARAMETRIC STUDY

A preliminary use of the completed program demonstrated was in the brief parametric study described below. The program solve.m (with input provided from a separate file test00.m) takes the converged result from the loop and computes a number of related parameters.

In order to assess the proximity to stall a very elementary approach is taken. The airfoil cascade is assumed to develop the deflection (lift) and drag provided by Howell's crude and early correlation of the cascaded airfoil. Thus the camber shape, stagger and solidity can be used to enter Howell's charts and predict the profile drag and lift (deflection) coefficients.

These data are developed for the representative section of single rotor compressor at the radius, or spanwise location, of the representative or characteristic section. For this study a design point wheel speed u=468 ft/sec, stagger angle β=45° and unit solidity are chosen* and various graphs of loading coefficient Ψ and rotor efficiency, η, can be plotted against flow coefficient, ϕ. These quantities derived from Howell's correlation are not presented as being quantitatively accurate. However, they do represent a consistent system of predicting drag-related performance para-meters, and the relative values are thought to indicate the
proximity to stall, or extent of separation. Thus, the rotor efficiency at the same flow coefficient is a measure of stall mitigation if the efficiency of the compliant rotor is higher than the efficiency of the rigid rotor.

In Fig. 4 and the charts following the results of an alternative study are shown for the rigid and compliant rotors having the following characteristics

	Rigid Blades	Compliant Blades
Unstrained Camber, deg.	22	15
Unstrained Stagger, deg.	45	45
Solidity	1.0	1.0
of Elasticity	1.5×10^7psi	1.0×10^6psi
Design wheel speed, fps	468	468

In this study the compliant and rigid blades have been tailored so as to have virtually the same performance at the wheel speed, u=468 fps. There the system is assumed to have an important second operating condition at 10% overspeed, or U=515 fps. The performance at this second condition then discriminates between the two rotors vis a vis stall mitigation.

Since the aerodynamics formulation is inviscid in the interests of rapidity of computation, an alternative assessment of the propensity to stall is made on the basis of several measures: incidence and a leading edge loading parameter taken in conjunction with the camber angle. The last is expected to change significantly with changing operating conditions for the compliant blade.

(lower) for the compliant rotor at all flow coefficients in the overspeed condition. It should be recalled that both rigid and compliant rotors have been configured to have virtually identical geometry and performance at design speed.

The difference shown in Fig. 4 has not been optimized, or maximized. By reducing the rigidity of the compliant blades still further, and increasing their unstrained, or "as fabricated", camber, the leading edge loading differences noted in the figure could be magnified. However, this optimization process has not been carried forward at the time of completing this Final Report. Only the potential of the method has been demonstrated.

REFERENCES

Sisto, F., "Linearized Theory of Nonstationary Cascades at Fully Stalled or Supercavitated Conditions", Zeitschrift fur Angewandte Mathematik und Physik, pp. 531-541, 1967.

Howell, A.R., "Fluid Dynamics of Axial Compressors", Proc. Instn. Mech. Engrs., No. 153, p. 445, 1945

APPENDIX A

CASCADE AERODYNAMICS

Initially we wish to derive the steady aerodynamic load on a cascade of camberlines in incompressible flow. The usual thin airfoil assumption is made that the vorticities (representing the jump in tangential velocity across the airfoils) are distributed along the chordlines and the boundary condition are satisfied along these same straight segments.

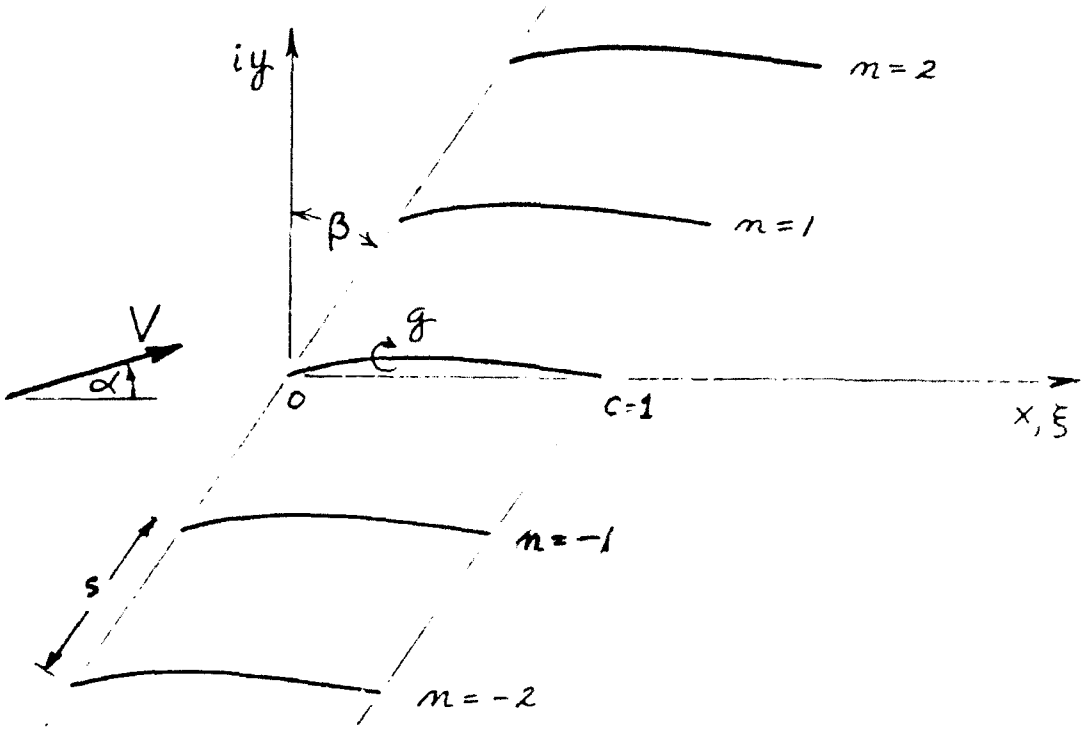


Fig. 1 Cascade Geometry

On the x-axis the velocities induced by the individual airfoils are given by*

$$u_0 - i v_0 = \frac{1}{2\pi i} \int_0^c g(\xi) \frac{d\xi}{\xi - x} \pm \frac{1}{2} g(x)$$

$$u_1 - i v_1 = \frac{1}{2\pi i} \int_{\Delta}^{\Delta + c} g(\xi_1) \frac{d\xi_1}{\xi_1 - x}$$

*The Biot-Savart Law expresses the conjugate velocity u-iv at a general field point. Subsequent applications of the Plemelj formulae supply the limiting values of u-iv as the camberline is approached from above (+) or below (-).

where $\xi_1 = \xi + \Delta$ and $\Delta = s e^{i(\frac{1}{2}\pi - \beta)}$ is the complex pitch, and so on for each airfoil.

Summing the contribution of all the (infinite number of) airfoils results in

$$u(x) - iv(x) = \pm\tfrac{1}{2} g(x) + \frac{1}{2\pi i} \sum_{n=-\infty}^{+\infty} \int_{n\Delta}^{n\Delta + c} g(\xi_n) \frac{d\xi_n}{\xi_n - x}, \qquad \xi_n = \xi + n\Delta$$

$$= \pm\tfrac{1}{2} g(x) + \frac{1}{2\pi i} \sum_{n=-\infty}^{+\infty} \int_{0}^{c} g(\xi) \frac{d\xi}{\xi + n\Delta - x} \qquad \because d\xi_n = d\xi$$

$$= \pm\tfrac{1}{2} g(x) + \frac{1}{2\pi i} \int_0^c g(\xi) \left[\frac{\pi e^{i\beta}}{s} \sum_{n=-\infty}^{+\infty} \frac{1}{\frac{\pi e^{i\beta}}{s}(\xi - x) + in\pi} \right] d\xi$$

Now a well-known result is $\sum_{n=-\infty}^{+\infty} [\chi + in\pi]^{-1} = \coth \chi$ so that

$$u(x) - iv(x) = \pm\tfrac{1}{2} g(x) + \frac{1}{2\pi i} \int_0^1 g(\xi') \left\{ P \coth [P(\xi' - x')] \right\} d\xi'$$

where

$$x' = \frac{x}{c}, \qquad \xi' = \frac{\xi}{c}.$$

Dropping the prime notation this may be put in the standard form

$$u(x) - iv(x) = \pm\tfrac{1}{2} g(x) + \frac{1}{2\pi i} \oint_0^1 g(\xi) \frac{d\xi}{\xi - x} + \frac{1}{2\pi i} \int_0^1 g(\xi) \left[F(\xi - x) + i\, G(\xi - x) \right] d\xi \qquad (1)$$

where $P = \pi \exp(i\beta)\, c/s$ (2a)

$$F(\xi - x; P) + i\, G(\xi - x; P) \equiv P \coth [P(\xi - x)] - \frac{1}{\xi - x} \qquad (2b)$$

and $F(0) = 0$, $G(0) = 0$

Since g is purely real, separation of Eq. (2) into real and imaginary parts yields

$$u(x) = \pm \tfrac{1}{2} g(x) + \frac{1}{2\pi} \int_0^1 g(\xi)\, G(\xi - x)\, d\xi \qquad (3)$$

$$v(x) = \frac{1}{2\pi} \oint_0^1 g(\xi) \frac{d\xi}{\xi - x} + \frac{1}{2\pi} \int_0^1 g(\xi) F(\xi - x)\, d\xi \qquad (4)$$

The boundary condition expresses the fact that the flow direction must be tangential to the blade surface (camberline in this case):

$$\frac{dy}{dx} = \frac{v + V \sin\alpha}{u + V \cos\alpha} \qquad (5)$$

where V and α refer to the mean flow and are defined in the figure*. Introduction of Equations (3) and (4) in Equation (5) results in the following integral equation

$$V\left(-\sin\alpha + \cos\alpha \frac{dy}{dx}\right) + \frac{1}{2\pi} \int_0^1 g(\xi) \left[-F(\xi - x) + \frac{dy}{dx} G(\xi - x)\right] d\xi = \frac{1}{2\pi} \oint_0^1 g(\xi) \frac{d\xi}{\xi - x} \qquad (6)$$

Solution of Integral Equation

If the LHS of this equation is presumed defined, then the inversion is well known, and is given by

$$g(x) = -\frac{2}{\pi} \sqrt{\frac{1-x}{x}} \oint_0^1 \sqrt{\frac{\xi}{1-\xi}} \left\{ V\left[-\sin\alpha + y'(\xi)\cos\alpha\right] + \frac{1}{2\pi} \oint_0^1 g(\eta)\left[-F(\eta - \xi) + y'(\xi) G(\eta - \xi)\right] d\eta \right\} \frac{d\xi}{\xi}$$

which may be written

$$g(x) = g_s(x) + \frac{1}{\pi} \sqrt{\frac{1-x}{x}} \oint_0^1 g(\eta)\, K(x, \eta)\, d\eta \qquad (7)$$

In the application of the B.C. the term $\pm \frac{1}{2} g(x)$ is omitted from $u(x)$ because of a fundamental approximation in thin airfoil theory*.

---

* Despite any local nonzero slope of the camberline the component u alone changes from one side of the boundary to the other. However, this jump in u is the local magnitude of the vorticity and physically the vorticity represents the jump in that component of the velocity which is parallel to the local boundary. Hence omitting the local vorticity (or strength of the vortex sheet) at the point where the B.C. are being applied, the remaining components of velocity must sum to a vector parallel to the local boundary. Another way of stating this condition is to say that if all components of the velocity, excepting the local vorticity, sum to a resultant which is parallel to the boundary, the subsequent addition of the vorticity component will not affect that parallelism (only the magnitude).

where g_s is the so-called single airfoil contribution

$$g_s \equiv \frac{2V}{\pi}\sqrt{\frac{1-x}{x}}\int_0^1 \sqrt{\frac{\xi}{1-\xi}}\left[\sin\alpha - y'(\xi)\cos\alpha\right]\frac{d\xi}{\xi - x} \tag{8}$$

and the kernel K is given by

$$K(x,\eta) \equiv -\frac{1}{\pi}\int_0^1 \sqrt{\frac{\xi}{1-\xi}}\left[-F(\eta-\xi) + y'(\xi)\,G(\eta-\xi)\right]\frac{d\xi}{\xi - x} \tag{9}$$

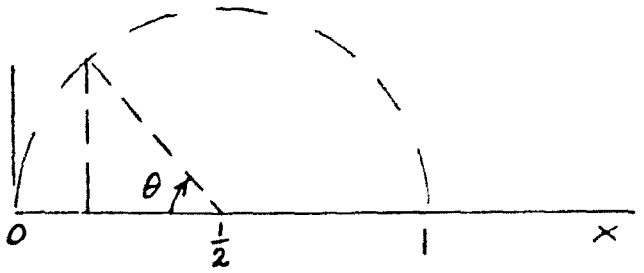
If a modified vorticity function, f, is defined by

$$Vf(x) \equiv g(x)\sqrt{\frac{x}{1-x}} \qquad Vf_s(x) \equiv g_s(x)\sqrt{\frac{x}{1-x}} \tag{10}$$

then the integral equation (7) takes the alternative form

$$f(x) = f_s(x) + \frac{1}{\pi}\int_0^1 \sqrt{\frac{1-\eta}{\eta}}\, f(\eta)\, K(x,\eta)\, d\eta \tag{11}$$

In order to obtain numerical solutions to this equation the following coordinate transformation, followed by discretization of the variables, is introduced.



$$x = \tfrac{1}{2}(1-\cos\theta) \qquad \theta = \frac{l\pi}{n} \tag{12a}$$

$$\eta = \tfrac{1}{2}(1-\cos\phi) \qquad \phi = \frac{\mu\pi}{n} \tag{12b}$$

$$\xi = \tfrac{1}{2}(1-\cos\psi) \qquad \psi = \frac{p\pi}{n} \tag{12c}$$

where l, μ and p are positive integers less than or equal to n, or else zero.

Under this parameterization Eqs. (8) and (9) reduce to

$$f_s(\theta) = -\frac{2}{\pi}\oint_0^{\pi}(1-\cos\psi)\left[\sin\alpha - y'(\psi)\cos\alpha\right]\frac{d\psi}{\cos\psi-\cos\theta} \qquad (8')$$

$$K(\theta,\phi) = \frac{1}{\pi}\oint_0^{\pi}(1-\cos\psi)\left[-F\left(\tfrac{1}{2}\cos\psi-\tfrac{1}{2}\cos\phi\right)+y'(\psi)G\left(\tfrac{1}{2}\cos\psi-\tfrac{1}{2}\cos\phi\right)\right]\times$$
$$\times\frac{d\psi}{\cos\psi-\cos\theta} \qquad (9')$$

and the integral appearing in Eq. (11) reduces to

$$\oint_0^1\sqrt{\frac{1-\eta}{\eta}}\,f(\eta)K(x,\eta)\,d\eta = \frac{1}{2}\oint_0^{\pi}(1+\cos\phi)\,f(\phi)K(\theta,\phi)\,d\phi \qquad (13)$$

If the numerators of the integrals in Eqs. (8') and (9') are assumed to be expanded in cosine series, it may be shown that these Glauert-type integrals may be approximated by sums

$$f_s(l) \cong -2\sum_{p=0}^{n}{}'' a_{lp}\left[1-\cos p\tfrac{\pi}{n}\right]\left[\sin\alpha - y'(p)\cos\alpha\right] \qquad (8'')$$

$$K(l,\mu) \cong \sum_{p=0}^{n}{}'' a_{lp}\left[1-\cos p\tfrac{\pi}{n}\right]\left[-F\left(\tfrac{1}{2}\cos p\tfrac{\pi}{n}-\tfrac{1}{2}\cos\mu\tfrac{\pi}{n}\right)+y'(p)\cdot\right.$$
$$\left.\times G\left(\tfrac{1}{2}\cos p\tfrac{\pi}{n}-\tfrac{1}{2}\cos\mu\tfrac{\pi}{n}\right)\right] \qquad (9'')$$

where

$$a_{lp} = \frac{1}{2}\left[2-\delta_{p0}-\delta_{pn}\right]\cdot\begin{cases}\dfrac{1}{n}\dfrac{1-(-1)^{l+p}}{\cos p\frac{\pi}{n}-\cos l\frac{\pi}{n}}, & p\neq l\\ 0, & p=l=1 \text{ or } n\\ n, & p=l=0\\ -n, & p=l=n\end{cases} \qquad (14)$$

and δ_{rs} is the Kronecker delta.

In Eq. (13), on the other hand, a direct application of the trapezoidal rule results in

$$\int_0^1 \sqrt{\frac{1-\eta}{\eta}}\, f(\eta) K(x,\eta)\, d\eta \cong \frac{\pi}{2n} \sum_{\mu=0}^{n} \left(1 - \tfrac{1}{2}\delta_{\mu 0} - \tfrac{1}{2}\delta_{\mu n}\right)\left(1 + \cos \mu\tfrac{\pi}{n}\right) f(\mu) K(\ell, \mu) \qquad (13'')$$

When these individual results are combined in integral equation (11) there results a set of $n+1$ simultaneous equations, each equation corresponding to a particular value of the index ℓ. These may be expressed in matrix notations as

$$\underline{A}\ \underline{F} = \underline{U} \qquad [A_{\ell\mu}]\{F_\ell\} = \{U_\ell\} \qquad (15)$$

where $[A_{\ell\mu}]$ is a $n+1$ square matrix in which ℓ identifies the row and μ the column. The $\ell\mu^{\text{th}}$ term is

$$A_{\ell\mu} = -\delta_{\mu\ell} + \frac{1}{4n}\left(2 - \delta_{\mu 0} - \delta_{\mu n}\right)\left(1 + \cos \mu\tfrac{\pi}{n}\right) \sum_{p=0}^{n} \eta_{\ell p} \left[1 - \cos p\tfrac{\pi}{n}\right] \times$$

$$\times \left[-F\left(\tfrac{1}{2}\cos p\tfrac{\pi}{n} - \tfrac{1}{2}\cos \mu\tfrac{\pi}{n}\right) + y'(p)\, G\left(\tfrac{1}{2}\cos p\tfrac{\pi}{n} - \tfrac{1}{2}\cos \mu\tfrac{\pi}{n}\right)\right] \qquad (16)$$

Both $\{F\}$ and $\{U\}$ are column matrices with $n+1$ rows in which the ℓ^{th} rows are given, respectively, by

$$F_\ell = f(\ell) \qquad (17)$$

$$U_\ell = 2 \sin\alpha \sum_{p=0}^{n} \eta_{\ell p} \left[1 - \cos p\tfrac{\pi}{n}\right] - 2\cos\alpha \sum_{p=0}^{n} \eta_{\ell p} \left[1 - \cos p\tfrac{\pi}{n}\right] y'(p) \qquad (18)$$

Partitioning $\underline{U}$ into two parts

$$U_\ell = \sin\alpha\, U_\ell^{(1)} - \cos\alpha\, U_\ell^{(2)} \tag{19}$$

where

$$U_\ell^{(1)} = 2\sum_{p=0}^{n} \mathcal{U}_{\ell p}\left[1-\cos\frac{p\pi}{n}\right] = -2 \tag{20}$$

$$U_\ell^{(2)} = 2\sum_{p=0}^{n} \mathcal{U}_{\ell p}\left[1-\cos\frac{p\pi}{n}\right] y'(p) \tag{21}$$

The solution of Eq. (15) may be expressed

$$\mathcal{F}_\ell = \sin\alpha\, \mathcal{F}_\ell^{(1)} - \cos\alpha\, \mathcal{F}_\ell^{(2)} \tag{22}$$

where

$$\underline{\mathcal{F}}^{(1)} = \underline{A}^{-1}\,\underline{U}^{(1)} \tag{23}$$

$$\underline{\mathcal{F}}^{(2)} = \underline{A}^{-1}\,\underline{U}^{(2)} \tag{24}$$

Cascade Properties

The circulation round one blade is

$$\Gamma = c\int_0^1 g(x)\,dx = cV\int_0^1 \sqrt{\frac{1-x}{x}}\, f(x)\,dx = cV\int_0^\pi \sqrt{\frac{1+\cos\theta}{1-\cos\theta}}\, f(\theta)\tfrac{1}{2}\sin\theta\,d\theta$$

$$\Gamma = \tfrac{1}{2} cV\int_0^\pi (1+\cos\theta)\, f(\theta)\,d\theta$$

$$\frac{\Gamma}{sV} \simeq \frac{\pi}{2n}\left(\frac{c}{s}\right)\sum_{\ell=0}^{n}\left(1-\tfrac{1}{2}\delta_{\ell 0}-\tfrac{1}{2}\delta_{\ell n}\right)\left(1+\cos\frac{\ell\pi}{n}\right) f(\ell)$$

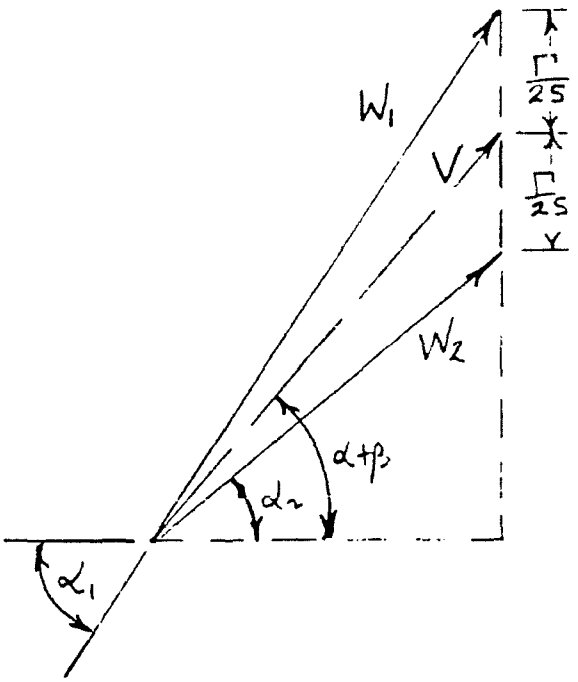
$$\frac{\Gamma}{sV} \simeq Q\sin\alpha - R\cos\alpha \tag{25}$$

where

$$Q = \frac{\pi}{2m}\left(\frac{c}{s}\right)\sum_{\ell=0}^{m}\left(1-\tfrac{1}{2}\delta_{\ell 0}-\tfrac{1}{2}\delta_{\ell m}\right)\left(1+\cos\frac{\ell\pi}{m}\right)A_{\ell}^{(1)} \qquad (26)$$

$$R = \frac{\pi}{2m}\left(\frac{c}{s}\right)\sum_{\ell=0}^{m}\left(1-\tfrac{1}{2}\delta_{\ell 0}-\tfrac{1}{2}\delta_{\ell m}\right)\left(1+\cos\frac{\ell\pi}{m}\right)A_{\ell}^{(2)} \qquad (27)$$

Knowing the circulation allows the mean flow velocity V and direction α to be related to the upstream (or downstream) resultant flow. Arranging the cascade along a vertical axis, the various velocity vectors take on the aspect shown in the sketch.



From the geometry of this sketch we deduce

$$-\tan\alpha_2 = \tan\alpha_1 - 2\tan(\alpha+\beta) \qquad (28)$$

whereas the dimensionless circulation for one pitch distance along the cascade is given by

$$\frac{\Gamma}{sV} = \cos(\alpha+\beta)\left[\tan\alpha_1 - \tan\alpha_2\right] \qquad (29)$$

Replacing $-\tan\alpha_2$ in Eq. (29) by Eq. (28) and then equating Eq. (25) to Eq. (29) results in

$$\tan\alpha = \frac{\frac{1}{2}R + \tan\alpha_1\cos\beta - \sin\beta}{\frac{1}{2}Q + \tan\alpha_1\sin\beta + \cos\beta} \qquad (30)$$

Presuming that α may be determined by this means, it is then possible to express V in terms of the physically determinate W_1

$$V = \frac{W_1 \cos \alpha_1}{\cos(\alpha+\beta)} = \frac{W_1 \cos \alpha_1 \sqrt{1+\tan^2 \alpha}}{\cos\beta - \sin\beta \tan\alpha} \qquad (31)$$

This results in the interesting conclusion that a given cascade may be solved up to the determination of R and Q without regard to the upstream flow conditions. Subsequently the velocity triangles may be deduced for any upstream flow condition W_1, α_1 by finding V, α from (30) and (31) and applying (28). In fact these manipulations may be used to derive Merchant's formula

$$\tan \alpha_2 = 2 \frac{R + Q \tan\beta}{2\sec\beta + Q - R\tan\beta} + \frac{2\sec\beta - Q + R\tan\beta}{2\sec\beta + Q - R\tan\beta} \tan\alpha_1 \qquad (32)$$

showing the linear dependence of $\tan\alpha_2$ on $\tan\alpha_1$.

APPENDIX B

PLATE MECHANICS

The moment acting on the spring supported plate, see Fig. 2, is expressed by first determining the aerodynamic moment about the leading edge

$$MOM_o = c\int_o^1 x\,dL = \tfrac{1}{2}\,qc^2\int_o^{\pi} f(\psi)\sin^2\psi\,d\psi \qquad (B1)$$

This allows the computation of the rearward spring force immediately from equilibrium considerations

$$f_r = MOM_o / x_{er} \qquad (B2)$$

Then computation of the aerodynamic lift

$$Lift = \int dL = qc\int_o^{\pi} f(\psi)\,(1+\cos\psi)\,d\psi \qquad (B3)$$

allows computation of the leading edge spring force

$$f_e = Lift - f_r$$

Knowing the spring constants kℓ and kr the leading edge and rearward support deflections may be computed in turn

$$Y(1,1) = f_e / k_e \qquad (B4)$$

$$Y(e_r,1) = f_r / k_r \qquad (B5)$$

where Y is the colum matrix (vector) of elastic displacement dimensionless w.r.t. the chord and the indices are given in terms of $1 \le$ indix $\le m \equiv n+1$.

The bending moment distribution may now be computed rearward from the trailing edge according to

$$M = \int_x^c (\xi - x)\,dL + U(e_r - e)\,f_r\,(x_r - x) \qquad (B6)$$

$$= \tfrac{1}{2}\,qc^2\int_o^{\pi}(1+\cos\psi)(\cos\theta-\cos\psi)f(\psi)\,d\psi + U(e_r-e)f_r\left(\frac{\cos\theta-\cos\theta_r}{2}\right)$$

where U is the unit step function and ξ, Ψ are dummy variables of integration analogous to x, θ.

In these integrals (B1, B6 and others) the integrals are determined at the discrete mesh points; hence the numerical integrations may be evaluated using either a trapezoidal rule, a Simpson's rule or other appropriate numerical technique.

For the approximation to the curvature $d^2ydx^2 = cd^2Y/dX^2$ several trial models were evaluated. Eventually, in order to perform the integration with uniform steps in the angular variable (θ or ψ) the following approximation was developed.

With Y≡y/c and X≡x/c=½(1-cos θ) it is found that

$$\frac{d^2y}{dx^2} = \frac{4}{c}\left[\sin\theta\,\frac{d^2Y}{d\theta^2} - \cos\theta\,\frac{dY}{d\theta}\right]\Big/\sin^3\theta$$

and expressing the derivatives in terms of control differences,

$$c/4\,h^3\frac{d^2y}{dx^2} = \mathcal{D} =$$

$$\left[h\sin(eu)\left(Y(e-1,1)-2Y(e,1)+Y(e+1,1)-\tfrac{1}{2}h^2\cos(eu)\left(-Y(e-1,1)+Y(e+1,1)\right)\right]\Big/\sin^3(eu)$$

where u=π/m and a common factor h is retained to reduce the size of the elements in the D matrix.

In the application of this result it is found that the first and last rows in the (n+1)x(n+1) matrix are not required since the bending moment is zero at leading and trailing edges. Recognizing that two values of Y(e,1) are known from previous equilibrium considerations, two columns may be removed and the useful D matrix may be reduced to (n-1)x(n-1) dimensions. In this reduced form the solution for the remaining Y terms may be obtained by inverting the reduced D matrix and solving for the remaining unknown values of Y

$$Y_r = inv(\mathcal{D}_r) * \left(\frac{M}{E}\right)_r$$

where the subscript r refers to the reduced versions of these matrices.

Upon reassembling the full Y vector it is then possible to estimate the elastic slopes of the camberline using central differences* to correct the stagger angle and the camberline slopes for the next iteration

$$\beta_{new}=\beta_{old}+\arctan(Y(m,1)-Y(1,1))$$

$$Y' \equiv Z$$

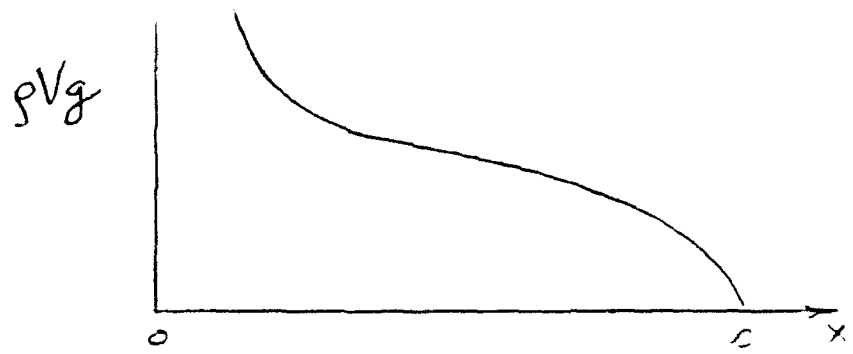
$$Z_{new}=\tan(\arctan Z_{old} + \arctan (Y(m,1)-Y(1,1))$$

Adding the elastic slopes to the unstrained camberline slopes the total slopes are then made available as inputs to the aerodyamics subroutine for the next iteration.

*For leading edge and trailing edge resp. forward and backward differences are used.

APPENDIX B

PLATE MECHANICS



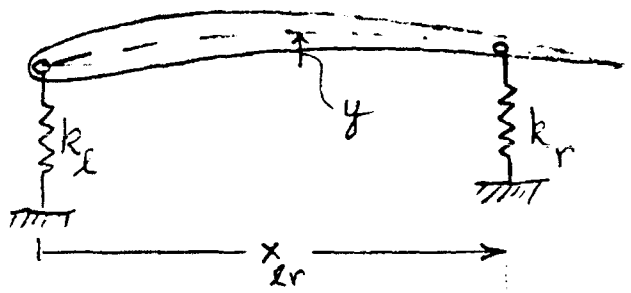


Fig. 2 Structural Model with Aerodynamic Load

APPENDIX C

Computer Programs

test 00.m

solve.m

in MatLab language

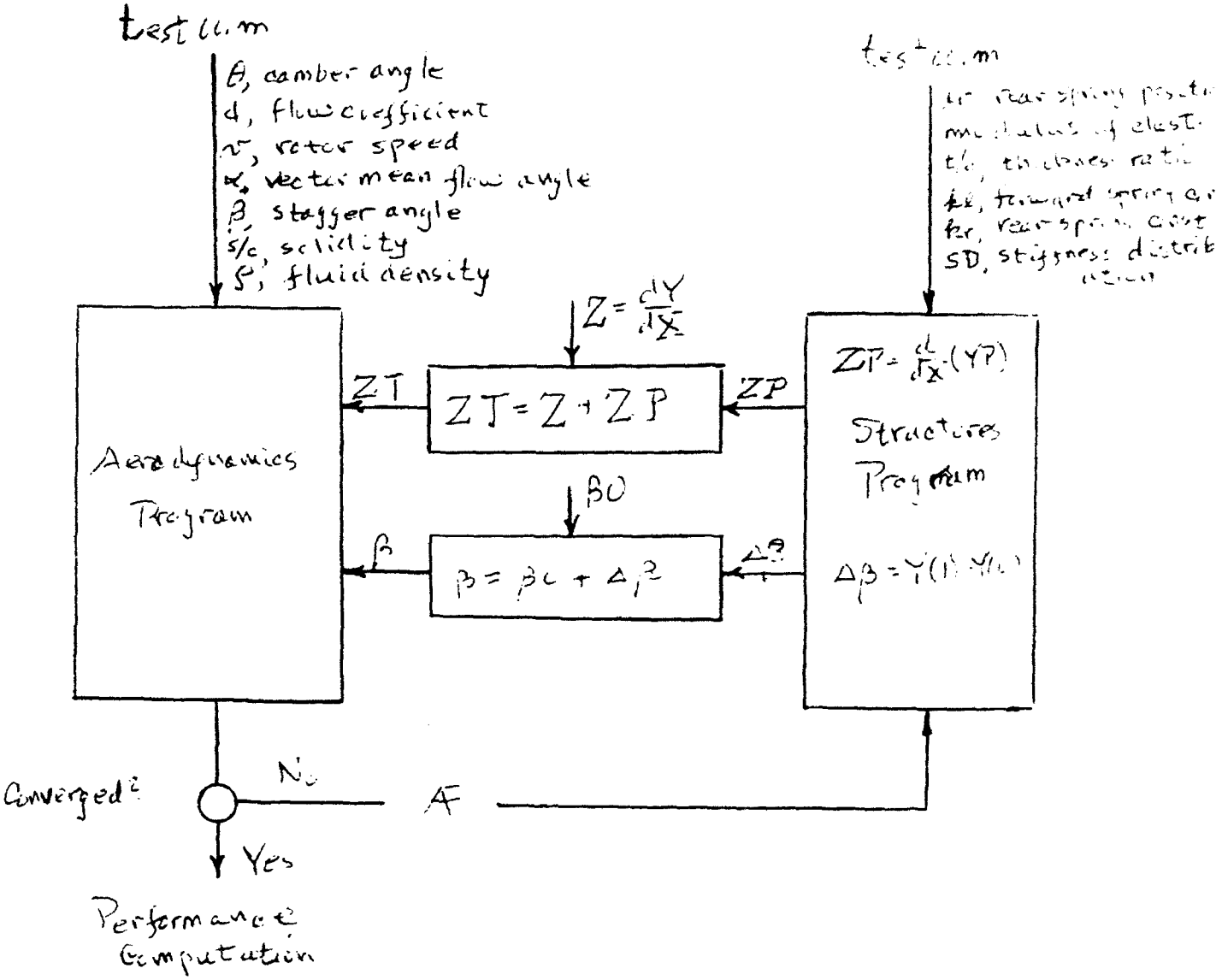


Fig. 3 Computation Flow Chart for solve.m Program (Matlab)

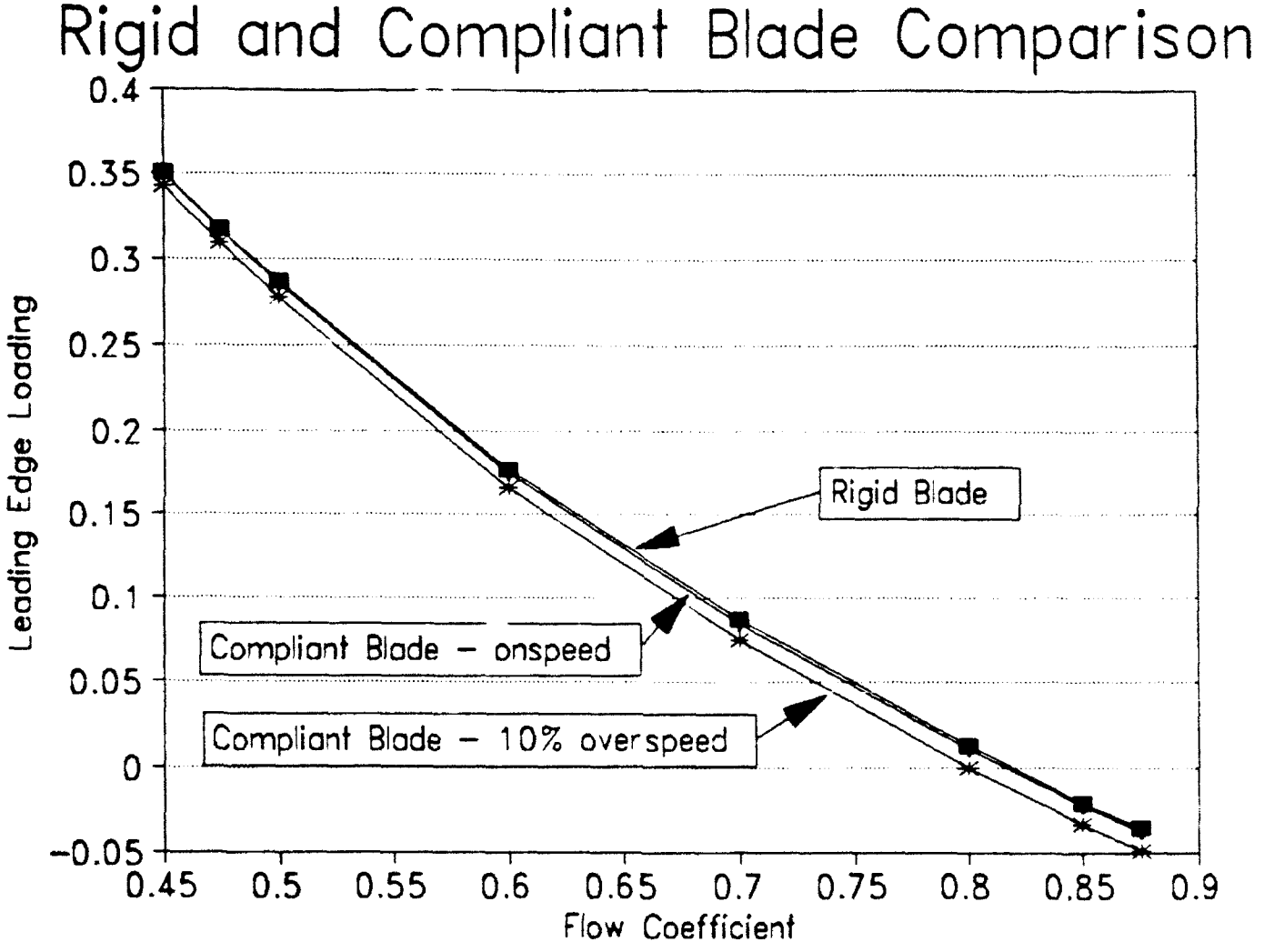


Fig. 4 Leading-Edge Loading Parameter for Rigid and Compliant-Bladed Rotors

```
%%%%%% SOLVE.M %%%%%%
%%%%%% CONSTANTS OUTSIDE OF ITERATION LOOPS %%%%%%%

dbeta = 1;                      % Initial dbeta to start iteration
dbetaold = 0;                   % Initial dbetaold for iteration
beta  = beta0;                  % Initial beta is the unstrained beta
alfa  = alfai;                  % Alfa to start iteration loop
alfaold  0;                     % Alfa to be replaced by alfai
n     = m - 1;                  % Number of panels
ZP    = zeros(m,1);             % Initial strained slopes
ZOLD  = ones(m,1);              % Old/previous strained slopes
count = 0;                      % Iteration counter
acount= 0;                      % Alfa counter
eps   = 0.0001*m;               % Iteration precision for slope summation
dif   = 1;                      % Iteration precision for flow coeff
h     = pi/n;                   % Angular segment interval
u       (cos(h)-cos(2*h))/(1-cos(h));

for a=1:m                       % Airfoil shape defined by circular arc
   X(a,1)   0.5*(1-cos((a-1)*h));
   YO(a,1)  -.5/tan(.5*theta)+ ((.5/(sin(.5*theta)))^2-(X(a,1)-.5)^2)^.5;
   Z(a,1)     (.5-X(a,1))/(YO(a,1)+.5/tan(.5*theta));
end

Z1      Z;                      % Initial total camber shape

for a 1:m                       % New stiffness matrix, m x m
   SD2(a,a)  (h^4*3)/SD(a,1)/tc3/2;
end

T       eye(m);                 % Trapezoidal matrix
T(1,1)  .5000;
T(m,m)  .5000;

SIMP(1,1) 2;                    % Simpson's 1/3 integration matrix
   cc 0;
for a 2:m
   if cc  0
      SIMP(1,a)  (1+cos((a-1)*h))*4;
      cc 1;
      else
      SIMP(1,a)  (1+cos((a-1)*h))^2;
      cc 0;
   end
end
SIMP SIMP/3;

for a 1:m                       % Generate the K matrix
   for b 1:m
      aa cos((a-1)*h);
      bb cos((b-1)*h);
      if a  b
         K(a,b) 0;
         else
         K(a,b)  (1-((1-1)^(a+b)))/(c*(bb-aa));
      end
   end
end
K(:,1) 0;
K(m,m) 0;
K       K';                     % K   product of K and trapezoidal matrix

SI zeros(m,m);                  % SI matrix
for a 1:m
   cc 0;
   for b 1:m
      sb (sin((b-1)*h))^2;
      cb 1+cos((b-1)*h);
      ca 1-cos((a-1)*h);
      if cc  0
         SI(a,b)  (sb+cb*ca)*4/3;
```



```
         cc=1;
         else
         SI(a,b)  (sb+cb*ca)*2/3;
         cc=0;
      end
   end
end
for a 2:2:(n-1)                 % Special Simpson,s  coefficient
   cc  0;
   for b (n-1):n
      sb=(sin((b-1)*h))^2;
      cb=1+cos((b-1)*h);
      ca=1-cos((a-1)*h);
      if cc  0
         SI(a,b)  (sb+cb*ca)*5/4;
         cc 1;
         else
         SI(a,b)  (sb+cb*ca);
         c  0;
      end
   end
end
a n-2;                          % Additional SI element
b n-1;
   sb (sin((b-1)*h))^2;
   cb 1+cos((b-1)*h);
   ca 1-cos((a-1)*h);
   SI(n,(n-1))  (sb+cb*ca)/12;

S2=zeros(m,m);                  % S2 matrix
for a 1:(ir-1)
   cc 0;
   for b 2:n
      sb (sin((b-1)*h))^2;
      ca cos((a-1)*h);
      clr cos(ir*h);
      if cc  0
         S2(a,b)  (ca-clr)*sb/(1-clr)*4/3;
         cc 1;
         else
         S2(a,b)  (ca-clr)*sb/(1-clr)*2/3;
         cc 0;
      end
   end
end

S   SI + S2;

D   zeros((m-2),m);             % D matrix
for a 2:n
   l a-1;
      D(l,(a-1))  h/(sin(l*h/2)^2)+.5*h^2*cos(l*h)/(sin(l*h)^3);
      D(l,a)  -2*h/(sin(l*h)^2);
      D(l,(a+1))  h/(sin(l*h)^2)-.5*h^2*cos(l*h)/(sin(l*h)^3);
   end
end
      dll D(1,1);               % Coefficient for leading edge force
      dllr D((ir-2),ir);        % Coefficients for rearward force
      d2ir D((ir-1),ir);
      d3r D(ir,ir);
D(:,[1 ir]) [];

%%%%% START ITERATION LOOP %%%%%

while abs(cc) > .0002

vel     velocity*cos(alfa+beta);        % Mean velocity for flowfield
q       0.5 * rho * vel^2;              % Dynamic pressure (psf)
qe      q*modulus;                      % Ratio of q/f
```

```
while abs(dbeta-dbetaold) > .01
while sum(abs(ZP-ZOLD)) > eps

dbetaold   dbeta;
count   count + 1
ZOLD   ZP;                    % switch old value before new ZP is solved

P    = pi*s*exp(beta*i);      % P constant

for a=1:m                     % F and G matrix formation
   for b=1:m
      aa=cos((a-1)*h);
      bb=cos((b-1)*h);
      if a==b
      H(a,b)=0;
         else
         H(a,b) P/tanh(0.5*P*(aa-bb))-2/(aa-bb);
      end
   end
end
F    = real(H);
G    = imag(H);

for a 1:m                     % Generate the upwash matrix
   sm 0;
   for b 1:m
      sm (K(a,b)*(1-cos((b-1)*h))*ZT(b,1))+sm;
   end
   U(a,1)=-2*(sin(alfa)+(cos(alfa)*sm));
end

for a=1:m                     % Generate aerodynamic matrix
   for b 1:m
      sm 0;
      for c 1:m
       sm K(a,c)*(1-cos((c-1.*E))*(F(b,c)-ZT(c,1)*G(b,c))+sm;
      end
   A(a,b) (1+cos((b-1)*h))*sm/4/n;
   sm 0;
   end
end
A A^1;
A A+eye(m);

FA   inv(A)*U;                % Aerodynamic force matrix

MOM   q*h*G1*FA;                        % Moment about the leading edge
fr     MOM(1,1)/(1-cos((lr-1)*h));      % Force on rear spring
lift   q*h*(SIMP*FA);                   % Lift generated by section
ff     lift - fr;                       % Force on forward spring

Y(1,1) ff/k;                            % LE deflection
Y(lr,1) fr/kr;                          % Rear Attachment def.

RHS   qo*SF2*(S*FA);

RHS(2,1) RHS(2,1)-d11*Y(1,1);
RHS((lr-1),1) RHS((lr-1),1)-d1lr*Y(lr,1);
RHS(lr,1) RHS(lr,1)-d21r*Y(1r,1);
RHS((lr+1),1) RHS((lr+1),1)-d31r*Y(lr,1);
RHS(( m),: )   (1);

YNEW    inv(D)*RHS;

for a 2:((lr-1)
   Y(a,1)   YNEW((a-1),1);
end
for a (lr+1):m
   Y(a,1)   YNEW((a-2),1);
end
```



```
dbeta = atan(Y(m,1)-Y(1,1));   % Delta beta

beta = beta0 + dbeta;          % New beta value
alfa = alfa1 - dbeta;          % New alfa value

ZP(1,1)=(2+u)*Y(1,1)/(1+u)-(1+u)*Y(2,1)/u+Y(3,1)/u/(1+u);
ZP(1,1)=-2*ZP(1,1)/(1-cos(h));
for a=2:m
   ZP(a,1)=(-Y((a-1),1)+Y((a+1),1))/h/sin((a-1)*h);
end
ZP(m,1) ((2+u)*Y(m,1)/(1+u)-(1+u)*Y((m-1),1)/u+Y((m-2),1)/u/(1+u));
ZP(m,1) 2*ZP(m,1)/(1-cos(h));

ZT    Z + ZP;                  % Total camber shape - unstrained and
                               % strained shapes superpositioned
for a 1:m
   ZT(a,1) tan(atan(ZT(a,1)-dbeta));
end

end                            % Close ZT Loop
end                            % Close dbeta Loop

circ     0.5*lift*o;           % Circulation Coeff
betal   atan(tan(alfa+beta)+0.5*circ/s/cos(alfa+beta));  % Relative flow
beta2   atan(tan(alfa+beta)-0.5*circ/s/cos(alfa+beta));  % angles
[ beta   [beta1];

account   account + 1
dif   flowcnt - flowno;

if abs(alfa1-alfa1old)< .00000001
   dif   0
end

if account= 1
if abs(dif)> .0050
   slope   (dif-difold)/(alfa1-alfa1old);
   alfa1   alfa1 - dif/slope;
   alfa1old   alfa1;
   alfa   alfa1 - dbeta;
   difold   dif;
   dbeta 1;
   dbetaold 0;
   ZP zeros(m,1);
   ZP(1,1)=zeros(m,1);
end
end

% account= 1
   alfa1old   alfa1;
   alfa1   alfa1+.02;
   alfa   alfa1-dbeta;
   dif=0 ;
   dbeta 1;
   flowcnt 0 ;
   ZP zeros(m,1);
   ZP(1,1)=zeros(m,1);
end

print alfa*180/pi

end                            % Close flowco loop

%%%%%  [ ] = flow type   %%%%%

gamma1   atan(dZ(1,1));        % leading edge camber
gamma2   atan(dZ(m,1));        % trailing edge camber
camber   gamma1   gamma2;      % Composite blade camber avg o

for a 1:m                      % Distribution of maximum camber
   Z Z(a,1)*2
```

```
285         ind =a;
286      end
287   end
288   maxcampt=ZT(ind,1)*(X((ind+1),1)-X(ind,1))/(ZT((ind+1),1)-ZT(ind,1));
289   maxcampt=X(ind,1)+maxcampt;
290
291   w1 = (1+sin(alfa+beta)*circ/s*((circ/s)^2)/4)^.5;    % Relative flow
292   w2 = (1-sin(alfa+beta)*circ/s*((circ/s)^2)/4)^.5;    % velocities
293
294   incid = beta1-beta-gamma1;                           % Incidence
295   delta  = beta2-beta-gamma2;                          % Deviation
296   nbeta2=(beta+gamma2)*180/pi/(s^.5)+0.23*(2*maxcampt)^2*camber*180/pi;
297   nbeta2=nbeta2/(1/(s^0.5)-camber*180/pi/500);
298
299   DEFTAB(1,:)=[0 .5 1.0 1.5 s];                        % Nominal deflection
300   DEFTAB(2,:)=[-10 52.5 39.5 31.7 0];
301   DEFTAB(3,:)=[0 43.3 33.0 26.8 0];
302   DEFTAB(4,:)=[10 36.8 27.9 22.75 0];
303   DEFTAB(5,:)=[30 26.25 19.7 16.2 0];
304   DEFTAB(6,:)=[50 16.4 12.5 10.2 0];
305   DEFTAB(7,:)=[70 7.75 6.15 4.75 0];
306   for c=2:7
307      for a=2:4
308         z=DEFTAB(c,a);
309         for b=2:4
310            if a~=b
311               z=z*(s-DEFTAB(1,b))/(DEFTAB(1,a)-DEFTAB(1,b));
312            end
313         end
314         DEFTAB(c,5)=DEFTAB(c,5)+z;
315      end
316   end
317   ndefl=0;
318   for a=2:7
319      z=DEFTAB(a,5);
320      for b=2:7
321         if a~=b
322            z=z*(nbeta2-DEFTAB(b,1))/(DEFTAB(a,1)-DEFTAB(b,1));
323          end
324      end
325      ndefl=ndefl+z;
326   end
327
328   % Nominal incidence (nincid) and incidence parameter (iparam)
329
330   nincid   nbeta2 + ndefl - (gamma2 + beta)*180/pi;
331   iparam   (incid*180/pi-nincid)/ndefl;
332
333   % Deflection parameter (eparam   defl/ndefl)
334
335   aa   -.8461;
336   bb    .6680;
337   cc   4.5421;
338   dd  -1.7093;
339   ee   2.3254;
340   ff   2.5899;
341   eparam   (1+aa*iparam+bb*(iparam^2)+cc*(iparam^3))
342   eparam   eparam/(1+dd*iparam+ee*(iparam^2)+ff*(iparam^3));
343
344   % profcd   profile drag coefficient
345
346   CDTAB(:,1) [-.6;-.5;-.4;-.3;-.2;-.1;0;.1;.2;.3;.4;.5;.6];
347   CDTAB(:,2) [.083;.05;.0315;.0244;.0192;.0168;.0168;.0174;.0192;.022;.046;.067;.121];
348   profcd=0;
349   for a=1:13
350      z=CDTAB(a,2);
351      for b=1:13
352         if a~=b
353            z=z*(iparam-CDTAB(b,1))/(CDTAB(a,1)-CDTAB(b,1));
354         end
355      end
```



```
356      profcd=profcd+z;
357   end
358
359   rbeta2 = beta1 - ndefl*eparam*pi/180;              % Correlation beta2
360   mbeta  = atan(.5*(tan(beta1)+tan(rbeta2)));        % Mean flow angle
361   alfa2  = atan(1/flowco-tan(rbeta2));               % Absolute angle
362   loadco   ) -flowco*tan(rbeta2);                    % Loading coefficient
363   liftco   2*s*cos(mbeta)*(tan(beta1)-tan(rbeta2));  % Lift coefficient
364   dragco = profcd + 0.009 + 0.018*(liftco^2);        % Drag coefficient
365   lossco   dragco/s/(cos(mbeta)^3);                  % Loss coefficient
366   roteff = 1- lossco*flowco^2/2/loadco;              % Rotor efficiency
367
368   ! rm diary                          % Delete the old diary file
369   diary                               % Open the diary file for output
370   count
371   dif                                 % flowcof - flowco
372   delalfai    (alfai - alfaiold)*180/pi
373   Z                                   % Unstrained slopes
374   ZT                                  % Total slopes (strained + unstrained)
375   camber   camber*180/pi
376   stagr = beta*180/pi
377   dbeta   dbeta*180/pi                % Change in Stagger due to strain
378   alfai   alfai*180/pi                % Mean flow w/o strain effect
379   alfa = alfa*180/pi                  % Mean flow angle
380   maxcampt                            % a/l = chordwise pointof max camber
381   beta1   beta1*180/pi                % Upstream relative flow angle
382   beta2   beta2*180/pi                % Computed downstream rel flow angle
383   incid   incid*180/pi                % Incidence
384   deviatn   delta*180/pi              % Deviation
385   nbeta2                              % Nominal downstream rel flow angle
386   ndefl                               % Nominal deflection
387   nincid                              % Nominal incidence
388   iparam                              % Incidence parameter
389   eparam                              % Deflection parameter (defl/ndefl)
390   profcd                              % Profile drag coefficient
391   rbeta2   rbeta2*180/pi              % Correlation downstream rel flow angle
392   alfa2   alfa2*180/pi                % Correlation downstream abs flow angle
393   flowco
394   loadco
395   liftco
396   dragco
397   lossco
398   roteff
399   l.f;
400   q
401   diary                               %Close the diary file
402
```

```
        % Aerodynamic input variables for circular arc unstrained camber
        % Uniform stiffness

theta   = 10*pi/180;          % Camber section angle (unstrained)
flowcof   .4500;              % Final flow coeff (target)
alfa0     11.4500*pi/180;     % Alfa to start iteration
beta0     40*pi/180;          % Unstrained stagger angle beta
s         1.00;               % s/c - solidity
m         13;                 % Number of nodes
rho       0.002377;           % STD density for air (slug/ft^3)
v         573.5;              % Rotor velocity (ft/sec)

        % Structural input variables

          11;                 % Position of rear spring (1<ir< m)
E         1.0e6;              % Modulus of elasticity
I         .12^3;              % (t/c)^3
Kf        24000;              % Forward spring constant
Kr        200;                % Rear spring constant
                              % Stiffness distribution at m points
S     [1.0;1.0;1.0;1.0;1.0;1.0;1.0;1.0;1.0;1.0;1.0;1.0;1.0];



        % Solve.m  should be run next
```

```
%%%%%% SOLVE.M %%%%%%
%%%%%% CONSTANTS OUTSIDE OF ITERATION LOOPS %%%%%%%

dbeta = 1;                          % Initial dbeta to start iteration
dbetaold = 0;                       % Initial dbetaold for iteration
beta    beta0;                      % Initial beta is the unstrained beta
alfa    alfa1;                      % Alfa to start iteration loop
alfaold = 0;                        % Alfa to be replaced by alfa1
m       = m - 1;                    % Number of panels
ZP      = zeros(m,1);               % Initial strained slopes
ZOLD    = ones(m,1);                % Old/previous strained slopes
count   = 0;                        % Iteration counter
acount  = 0;                        % Alfa counter
eps     = 0.0001*m;                 % Iteration precision for slope summation
dif     = 1;                        % Iteration precision for flow coeff
h       = pi/n;                     % Angular segment interval
c       = (cos(h)-cos(2*h))/(1-cos(h));

for a=1:m                           % Airfoil shape defined by circular arc
   X(a,1)   = 0.5*(1-cos((a-1)*h));
   YO(a,1)  = -.5/tan(.5*theta)+ ((.5/(sin(.5*theta)))^2-(X(a,1)-.5)^2)^.5;
   Z(a,1)   = (.5-X(a,1))/(YO(a,1)+.5/tan(.5*theta));
end

ZT      Z;                          % Initial total camber shape

for a 1:m                           % New stiffness matrix, m x m
   SD2(a,a)=(h^4*3)/SD(a,1)/tc)/2;
end

I       = eye(m);                   % Trapezoidal matrix
T(1,1) = .5000;
T(m,m) = .5000;

SIMP(1,1)=2;                        % Simpson's 1/3 integration matrix
   cc=0;
for a=2:m
   if cc=0
      SIMP(1,a)=(1+cos((a-1)*h))*4;
      cc=1;
      else
      SIMP(1,a)=(1+cos((a-1)*h))*2;
      cc=0;
   end
end
SIMP=SIMP/3;

for a=1:m                           % Generate the K matrix
   for b=1:m
      aa=cos((a-1)*h);
      bb=cos((b-1)*h);
      if a=b
         K(a,b)=0;
         else
         K(a,b)=(1-((-1)^(a+b)))/(n*(bb-aa));
      end
   end
end
K(1,1)=0;
K(m,m)=0;
KT       K*T;                       % KT = product of K and trapezoidal matrix

S1=zeros(m,m);                      % S1 matrix
for a=1:m
   cc=0;
   for b=(a+1):n
      sb=(sin((b-1)*h))^2;
      cb=1+cos((b-1)*h);
      ca=1-cos((a-1)*h);
      if cc=0
         S1(a,b)=(sb-cb*ca)*4/3;
```



```
            cc=1;
            else
            S1(a,b)=(sb-cb*ca)*2/3;
            cc=0;
         end
      end
   end
   for a=2:2:(n-1)                  % Special Simpson,s  coefficient
      cc=0;
      for b=(n-1):n
         sb=(sin((b-1)*h))^2;
         cb=1+cos((b-1)*h);
         ca=1-cos((a-1)*h);
         if cc=0
            S1(a,b)=(sb-cb*ca)*5/4;
            cc=1;
            else
            S1(a,b)=(sb-cb*ca);
            cc=0;
         end
      end
   end
   a n-2;                           % Additional S1 element
   b n-1;
      sb=(sin((b-1)*h))^2;
      cb 1+cos((b-1)*h);
      ca=1-cos((a-1)*h);
      S1(n,(n-1))=-(sb-cb*ca)/12;

   S2=zeros(m,m),                   % S2 matrix
   for r=1:(lr-1)
      cc=0;
      for b=2:n
         sb=(sin((b-1)*h))^2;
         ca=cos((a-1)*h);
         clr=cos(lr*h);
         if cc=0
            S2(a,b)=(ca-clr)*sb/(1-clr)*4/3;
            cc=1;
            else
            S2(a,b)=(ca-clr)*sb/(1-clr)*2/3;
            cc=0;
         end
      end
   end

   S   S1 - S2;

   D   zeros((m-2),m);              % D matrix
   for a=2:n
      i=a-1;
         D(i,(a-1))=h/(sin(i*h)^2)+.5*h^2*cos(i*h)/(sin(i*h)^3);
         D(i,a) -2*h/(sin(i*h)^2);
         D(i,(a+1)) h/(sin(i*h)^2)-.5*h^2*cos(i*h)/(sin(i*h)^3);
      end
   end
         d11=D(1,1);                % Coefficient for leading edge force
         d1lr=D((lr-2),lr);         % Coefficients for rearward force
         d2lr=D((lr-1),lr);
         d3lr=D(lr,lr);
   D(:,[1 lr]) [];

   %%%%% START ITERATION LOOP  %%%%%


   while abs(dif) > .0001

   vel     v*flowcof/cos(alfa+beta);      % Mean velocity for flowcof
   q       0.5 * rho * vel^2;             % Dynamic pressure (psf)
   qe      q/modulus;                     % Ratio of q/E
```

```
43   while abs(dbeta-dbetaold) > .01
44   while sum(abs(ZP-ZOLD)) > eps
45
46   dbetaold = dbeta;
47   count =count + 1
48   ZOLD  = ZP;                    % switch old value before new ZP is solved
49
50   P    = pi*s*exp(beta*i);       % P constant
51
52   for a=1:m                      % F and G matrix formation
53      for b=1:m
54         aa=cos((a-1)*h);
5          bb=cos((b-1)*h);
6          if a==b
7          H(a,b)=0;
8             else
9             H(a,b)=P/tanh(0.5*P*(aa-bb))-2/(aa-bb);
10         end
11      end
12   end
13   F    = real(H);
14   G      imag(H);
15
16   for a=1:m                      % Generate the upwash matrix
17      sm=0;
18      for b=1:m
19         sm=(K(a,b)*(1-cos((b-1)*h)))*ZT(b,1))+sm;
         end
         U(a,1)=-2*(sin(alfa)+(cos(alfa)*sm));
2    end
3
4    for a 1:m                      % Generate aerodynamic matrix
        for b 1:m
6          sm=0;
7          for c 1:m
8           sm K(a,c)*(1-cos((c-1)*h)))*(F(b,c)-ZT(c,1)*G(b,c))+sm;
9          end
10      A(a,b)=(1+cos((b-1)*h))*sm/4/n;
11      sm=0;
12      end
13   end
14   A A*1;
15   A A-eye(m);
16
17   FA   inv(A)*U;                 % Aerodynamic force matrix
18
19   MOM = q*h*S1*FA;                          % Moment about the leading edge
20   fr   = MOM(1,1)/(1-cos((lr-1)*h));        % Force on rear spring
21   lift = q*h*(SIMP*FA);                     % Lift generated by section
22   fl   = lift - fr;                         % Force on forward spring
23
24   Y(1,1)=fl/kl;                             % LE deflection
25   Y(lr,1)=fr/kr;                            % Rear Attachment defl
26
27   RHS = qe*SD2*(S*FA);
28
29   RHS(2,1)=RHS(2,1)-d11*Y(1,1);
   RHS((lr-1),1)=RHS((lr-1),1)-d1lr*Y(lr,1);
   RHS(lr,1)=RHS(lr,1)-d21r*Y(lr,1);
2  RHS((lr+1),1)=RHS((lr+1),1)-d31r*Y(lr,1);
3  RHS(3:m,1)=0;
4
   YNEW    inv(D)*RHS;
6
7  for a 2:(lr-1)
8     Y(a,1)    YNEW((a-1),1);
9  end
   for a (lr+1):m
      Y(a,1)    YNEW((a-2),1);
2  end
3
```



```
214  dbeta = atan(Y(m,1)-Y(1,1));   % Delta beta
215
216  beta = beta0 + dbeta;          % New beta value
217  alfa = alfai - dbeta;          % New alfa value
218
219  ZP(1,1)=(2+u)*Y(1,1)/(1+u)-(1+u)*Y(2,1)/u+Y(3,1)/u/(1+u);
220  ZP(1,1)=-2*ZP(1,1)/(1-cos(h));
221  for a=2:m
222     ZP(a,1)=(-Y((a-1),1)+Y((a+1),1))/h/sin((a-1)*h);
223  end
224  ZP(m,1)=((2+u)*Y(m,1)/(1+u)-(1+u)*Y((m-1),1)/u+Y((m-2),1)/u/(1+u));
225  ZP(m,1)=2*ZP(m,1)/(1-cos(h));
226
227  ZT    = Z + ZP;                % Total camber shape - unstrained and
228                                 % strained shapes superpositioned
229  for a=1:m
230     ZT(a,1)=tan(atan(ZT(a,1)-dbeta));
231  end
232
233  end                            % Close ZT Loop
234  end                            % Close dbeta Loop
235
236  circ    = 0.5*lift/q;          % Circulation Coeff
237  beta1 = atan(tan(alfa+beta)+0.5*circ/s/cos(alfa+beta));  % Relative flow
238  beta2 = atan(tan(alfa+beta)-0.5*circ/s/cos(alfa+beta));  % angles
239  flowco = 1/tan(beta1);
240
241  acount = acount + 1
242  dif = flowcof - flowco;
243
244  if abs(alfai-alfaiold)< .00000001
245     dif = 0
246  end
247
248  if acount==1
249  if abs(dif)>=.0050
250     slope = (dif-difold)/(alfai-alfaiold);
251     alfai = alfai - dif/slope;
252     alfaiold = alfai;
253     alfa = alfai - dbeta;
254     difold = dif;
255     dbeta 1;
256     dbetaold 0;
257     ZP zeros(m,1);
258     ZOLD ones(m,1);
259  end
260  end
261
262  if acount  1
263     alfaiold = alfai;
264     alfai  alfai-.02;
265     alfa = alfai-dbeta;
266     difold=dif;
267     dbeta=1;
268     dbetaold=0;
269     ZP zeros(m,1);
270     ZOLD ones(m,1);
271  end
272
273  print alfai*180/pi
274
275  end                            % Close flowco loop
276
277  $$$$$  OUTPUT FROM LOOP   $$$$$
278
279  gamma1   atan(ZT(1,1));        % Leading edge camber
280  gamma2   atan(ZT(m,1));        % Trailing edge camber
281  camber   gamma1 - gamma2;      % Compliant blade camber angle
282
283  for a 1:m                      % Position of maximum camber
284     if ZT(a,1)> 0
```

```
285         ind =a;
286      end
287   end
288   maxcampt=ZT(ind,1)*(X((ind+1),1)-X(ind,1))/(ZT((ind+1),1)-ZT(ind,1));
289   maxcampt=X(ind,1)-maxcampt;
290
291   w1 = (1+sin(alfa+beta)*circ/s+((circ/s)^2)/4)^.5;   % Relative flow
292   w2 = (1-sin(alfa+beta)*circ/s+((circ/s)^2)/4)^.5;   % velocities
293
294   incid = beta1-beta-gamma1;                          % Incidence
295   delta  = beta2-beta-gamma2;                         % Deviation
296   nbeta2=(beta+gamma2)*180/pi/(s^.5)+0.23*(2*maxcampt)^2*camber*180/pi;
297   nbeta2=nbeta2/(1/(s^0.5)-camber*180/pi/500);
298
299   DEFTAB(1,:)=[0 .5 1.0 1.5 s];                       % Nominal deflection
300   DEFTAB(2,:)=[-10 52.5 39.5 31.7 0];
301   DEFTAB(3,:)=[0 43.3 33.0 26.8 0];
302   DEFTAB(4,:)=[10 36.8 27.9 22.75 0];
303   DEFTAB(5,:)=[30 26.25 19.7 16.2 0];
304   DEFTAB(6,:)=[50 16.4 12.5 10.2 0];
305   DEFTAB(7,:)=[70 7.75 6.15 4.75 0];
306   for c=2:7
307      for a=2:4
308         z=DEFTAB(c,a);
309         for b 2:4
310            if a~= b
311               z=z*(s-DEFTAB(1,b))/(DEFTAB(1,a)-DEFTAB(1,b));
312            end
313         end
314         DEFTAB(c,5)=DEFTAB(c,5)+z;
315      end
316   end
317   ndef1=0;
318   for a=2:7
319      z=DEFTAB(a,5);
320      for b=2:7
321         if a~=b
322            z=z*(nbeta2-DEFTAB(b,1))/(DEFTAB(a,1)-DEFTAB(b,1));
323          end
324      end
325      ndef1=ndef1+z;
326   end
327
328   % Nominal Incidence (nincid) and Incidence parameter (iparam)
329
330   nincid = nbeta2 + ndef1 - (gamma1 + beta)*180/pi;
331   iparam  =(incid*180/pi-nincid)/ndef1;
332
333   % Deflection parameter (eparam = def1/ndef1)
334
335   aa   =-.8461;
336   bb =  .6680;
337   cc   4.5401;
338   dd  -1.7093;
339   ee   2.3254;
340   ff   2.5890;
341   eparam = (1+aa*iparam+bb*(iparam^2)+cc*(iparam^3))
342   eparam = eparam/(1+dd*iparam+ee*(iparam^2)+ff*(iparam^3));
343
344   % profcd   profile drag coefficient
345
346   CDTAB(:,1) =[-.6;-.5;-.4;-.3;-.2;-.1;0;.1;.2;.3;.4;.5;.6];
347   CDTAB(:,2) =[.083;.05;.0335;.0234;.0182;.0168;.0168;.0173;.0182;.0225;.036;.067;.123];
348   profcd=0;
349   for a 1:13
350      z CDTAB(a,2);
351      for b 1:13
352         if a~=b
353            z z*(iparam-CDTAB(b,1))/(CDTAB(a,1)-CDTAB(b,1));
354         end
355      end
```



```
356      profcd=profcd+z;
357   end
358
359   rbeta2 = beta1 - ndef1*eparam*pi/180;              % Correlation beta2
360   mbeta  = atan(.5*(tan(beta1)+tan(rbeta2)));        % Mean flow angle
361   alfa2  = atan(1/flowco-tan(rbeta2));               % Absolute angle
362   loadco = 1 -flowco*tan(rbeta2);                    % Loading coefficient
363   liftco = 2*s*cos(mbeta)*(tan(beta1)-tan(rbeta2));  % Lift coefficient
364   dragco = profcd + 0.009 + 0.018*(liftco^2);        % Drag coefficient
365   lossco = dragco/s/(cos(mbeta)^3);                  % Loss coefficient
366   roteff = 1- lossco*flowco^2/2/loadco;              % Rotor efficiency
367
368   ! rm diary                          % Delete the old diary file
369   diary                               % Open the diary file for output
370   count
371   dif                                 % flowcof - flowco
372   delalfai  = (alfai - alfaiold)*180/pi
373   Z                                   % Unstrained slopes
374   ZT                                  % Total slopes (strained + unstrained)
375   camber = camber*180/pi
376   stagr = beta*180/pi
377   dbeta = dbeta*180/pi                % Change in Stagger due to strain
378   alfai = alfai*180/pi                % Mean flow w/o strain effect
379   alfa  alfa*180/pi                   % Mean flow angle
380   maxcampt                            % a/l = chordwise pointof max camber
381   beta1 = beta1*180/pi                % Upstream relative flow angle
382   beta2 = beta2*180/pi                % Computed downstream rel flow angle
383   incid = incid*180/pi                % Incidence
384   deviatn = delta*180/pi              % Deviation
385   nbeta2                              % Nominal downstream rel flow angle
386   ndef1                               % Nominal deflection
387   nincid                              % Nominal incidence
388   iparam                              % Incidence parameter
389   eparam                              % Deflection parameter (def1/ndef1)
390   profcd                              % Profile drag coefficient
391   rbeta2 = rbeta2*180/pi              % Correlation downstream rel flow angle
392   alfa2 = alfa2*180/pi                % Correlation downstream abs flow angle
393   flowco
394   loadco
395   liftco
396   dragco
397   lossco
398   roteff
399   lift
400   q
401   ylead = Y(1,1)
402   ytrail = Y(ir,1)
403   YNEW
404   Y
405   diary                               %Close the diary file
406
```

DISTRIBUTION LIST

Director (2)
Defense Tech Information Center
Cameron Station
Alexandria, VA 22314

Research Office (1)
Code 81
Naval Postgraduate School
Monterey, CA 93943

Library (2)
Code 52
Naval postgraduate School
Monterey, CA 93943

Professor Max F. Platzer (5)
Department of Aeronautics and
Astronautics
Naval Postgraduate School
Monterey, CA 93943

Professor F. Sisto (5)
Department of Mechanical Engineering
Stevens Institute of Technology
Hoboken, NJ 07030-5991